Automatización en la Reconfiguración y Recuperación de un Cluster por Fallas de Nodos

Maximiliano Crescitelli, Emmanuel Bello, Julio Monetti, Oscar Leon
Fernando G. Tinetti[1]
Universidad Tecnológica Nacional - FRMendoza
Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática - UNLP

Resumen

El trabajo se centra en definir aquellas características sobresalientes de un cluster de computadoras, y cuáles son necesarias para la configuración del mismo. Específicamente, se analizan las posibles variables a tener en cuenta cuando el nodo encargado de la sincronización de tareas y políticas generales de uso (en adelante *nodo master*) sale de servicio e imposibilita la ejecución de los procesos.

Palabras Claves: Cluster, Computación Paralela, Tolerancia a Fallas.

Contexto

Esta línea de Investigación surge como interés de alumnos de Ingeniería en Sistemas de Información, bajo la asistencia de docentes e investigadores en la línea de computación paralela.

Introducción

Un cluster de computadoras es utilizado generalmente en el mundo académico y científico para la ejecución de sistemas que utilizan grandes volúmenes de información o demandan una gran cantidad de procesamiento. Al utilizarse éstos dentro del ámbito académico, no se encuentran generalmente soluciones comerciales que permitan atender problemas puntuales. Debido a esto, cada administrador de cluster se ve obligado a realizar tareas de configuración específicas sobre su arquitectura. Se trabaja actualmente en el análisis de una solución que permita la recuperación de un cluster ante la salida de servicio de alguno de sus nodos, especialmente el *master* [1]. Entre las diferentes soluciones se analiza la utilización de virtualización de los servicios que permiten el funcionamiento del mismo.

En el presente trabajo se atiende el problema mencionado a través de la reconfiguración de nodos en forma automática, intentando el cambio de roles de un nodo *worker* para convertirse en nodo *master*, ante la ausencia de este último.

Se exponen las características de la instalación de un cluster, analizando los principales elementos de software involucrados en dicha instalación. Luego, se analizan los problemas más frecuentes en el uso de la nueva plataforma, y más precisamente aquellos experimentados por el grupo de investigación. A partir de dichos problemas se analiza una propuesta de solución a la salida de servicio de un nodo, y se mencionan los resultados de los experimentos realizados, en base a la so-

[1] Investigador Comisión de Investigaciones Científicas de la prov. de Buenos Aires.

lución propuesta. Finalmente se presentan las conclusiones obtenidas a partir del análisis de los resultados de los experimentos.

Líneas de Investigación y Desarrollo

Elementos de configuración de un cluster

Una vez instalado el sistema operativo, se deben instalar y configurar servicios tendientes, entre otras cosas, a permitir la interacción entre nodos, la administración de seguridad en el acceso, la administración de un almacenamiento de datos común entre nodos, etc.

La gran mayoría de los clusters tienen la tendencia a identificar uno de los nodos como el nodo *master*. Dos de las razones más importantes son:

- La utilización de NFS [2] (*Network File System*) para evitar la replicación de ficheros, por ejemplo los ejecutables de un programa paralelo SPMD (*Single Program Multiple Data*).
- El registro (*login*) unificado, para evitar que un usuario tenga que registrarse individualmente en cada uno de los nodos del cluster.

Además de NFS se suelen utilizar protocolos como NIS (*Network Information System*) [2] o implementaciones de LDAP (*Lightweight Directory Access Protocol*) para resolver tareas de administración de usuarios de manera centralizada, en un nodo *master* o *server* del cluster.

A partir de esta necesidad surgen otras necesidades de compartir información de sistema en una red. Este inconveniente es solucionado a través de la utilización del protocolo NIS, utilizado este para compartir datos de configuración de sistema sobre una red. Este protocolo permite "exportar" y compartir información sensible sobre autenticación de usuarios al resto de los nodos, permitiendo así que un usuario registrado en el nodo *master* pueda ingresar con la misma clave en cada nodo donde se encuentre configurado el protocolo NIS (cliente).

Se debe tener en cuenta que si bien cada nodo del cluster forma parte de un todo, no deja de ser un sistema autónomo, el cual requiere autenticación y autorización de usuarios para hacer uso del mismo (generalmente por parte del *master*). Por la naturaleza del cómputo paralelo, la comunicación entre procesos residentes en los diferentes nodos se transforma en el aspecto central del funcionamiento de un cluster. Para la ejecución remota de procesos sobre los nodos del cluster, se ha estandarizado el uso del *ssh* (*Secure Shell*). Este protocolo permite satisfacer tanto cuestiones de seguridad como de comunicación. Todas las transferencias de información a través del canal de comunicación se realizan de forma encriptada, utilizando el sistema RSA [3] y evitando cualquier tipo de intercepción desde el medio. Para el correcto funcionamiento del protocolo, en tiempo de instalación del cluster, se generan ficheros conteniendo claves públicas y privadas, creando copias de las primeras a lo largo en los diferentes nodos.

Problemas frecuentes en el uso de un cluster

Básicamente los problemas experimentados, más allá de las fallas del algoritmo paralelo, o problemas puntuales en la configuración de los nodos, se pueden centrar en la salida de servicio de alguno de éstos. Si el nodo que falla es el *master*, el problema se acrecienta, puesto que en este nodo generalmente residen los procesos de sincronización, control, el directorio compartido con los ficheros de trabajo, etc.

Una forma muy económica de construir un cluster, consiste en aprovechar cualquier computadora o laboratorio que presente tiempos ociosos. Como en el presente trabajo el cluster utilizado para los experimentos corresponde a un laboratorio de usos generales, las probabilidades de desajustes y salidas de servicio de cada nodo son extremadamente altas, y de hecho peligrosas para cualquier usuario que necesite hacer un cómputo crítico sobre el cluster. Por lo tanto, si se cuenta con un alto riesgo de fallas, y con un cluster con un costo de construcción bajo, resulta conveniente proponer una solución referida a la reconfiguración automática del mismo. En consecuencia, en el presente trabajo se automatiza parte de la configuración del cluster y se propone una solución a la posible falla del *master*, la cual se denominará *Master Rotativo*.

Solución Propuesta

De acuerdo a los elementos señalados en la sección anterior, donde se mencionan los principales componentes de software necesarios para implementar un cluster, se analizan las diferentes soluciones en la implementación de metodologías tendientes a la recuperación de la arquitectura frente a la salida de servicio de un nodo.

La presente propuesta se aplica en función de dos aspectos: la *anatomía* y la *fisiología* de la solución. El primer aspecto tiene en cuenta las características estructurales del cluster, a través de sus protocolos, servicios y constitución física. De acuerdo a lo expresado anteriormente, la instalación del *master* difiere de la de cada nodo *worker*, puesto que en el primero residen los servicios encargados de la sincronización, autenticación y autorización de usuarios, entre otros. Este trabajo trata la reconfiguración de los protocolos NFS y NIS, cuya configuración inicial se diferencia según residan en el *master* o *worker*. Básicamente, la *anatomía* de la solución propone la reconfiguración automática de los ficheros del sistema, pertenecientes a los protocolos NFS, NIS y SSH. En el momento de iniciación de cada nodo, un programa analiza la situación actual del cluster, y de acuerdo a lo observado realiza la modificación de los mencionados ficheros, antes de que se inicien los protocolos en cuestión. Es tarea del programa individualizar y separar los ficheros de configuración en dos grupos: *ficheros universales* y *ficheros del sistema*. Se denominan ficheros universales a aquellos que no difieren en cuanto a su contenido según residan en el *master* o *workers*. Por otro lado, los ficheros del sistema son aquellos en los que sí difiere el contenido de acuerdo a la clasificación anterior. Como ejemplo de ficheros universales relacionados con el protocolo NIS se pueden mencionar el *yp.conf*, *defaultdomain*, *host.conf*, *nsswitch.conf*.

Como ficheros del sistema se pueden mencionar: *passwd*, *group*, *shadow*, *gshadow* (información de los usuarios, sus claves encriptadas, etc) y *fstab* (información de los medios de almacenamiento y puntos de montaje). Luego, es posible reemplazar un fichero de configuración en el momento del inicio del nodo sin alterar el funcionamiento autónomo del mismo. En otros casos, como en el tratamiento del *fstab*, el programa de inicio puede alterar la información en él contenida, de acuerdo a si se trabaja sobre un nodo *worker* o en el *master*. Se hace notar que, por razones de seguridad, si se desea reemplazar uno de los ficheros del sistema, éste debería ser editado por el programa de inicio, pero no podría ser reemplazado por uno de otro nodo, ya que generalmente cuenta con información sensible y utilizada por otras aplicaciones. Cada nodo contiene un directorio que cuenta con ficheros de configuración

auxiliares (copias de ficheros de configuración de NIS, NFS, SSH), los cuales son administrados a discreción del programa de iniciación (un detalle de esta administración se puede encontrar en [4]). Como los ficheros universales pueden copiarse reemplazando el ya existente, se realiza esta acción en una primera instancia con el objeto de comenzar a configurar el *worker* como *master*.

Habiendo seguido este razonamiento, desde el punto de vista de la *anatomía*, la solución propuesta se materializa en el programa de inicialización, el cual es ejecutado en etapas tempranas del proceso de inicio de cada nodo, con el objeto de que los protocolos encargados del funcionamiento del cluster puedan ver reflejados estos cambios en el momento de comienzo de la ejecución de los mismos.

La fisiología de la solución tiene en cuenta el comportamiento de los nodos al momento de observar la ausencia del nodo *master* en el momento de iniciación. Esta parte de la solución se materializa en el protocolo ISC [5], el cual trabaja bajo la premisa de que cada nodo debe proveer la posibilidad de tomar el rol de *master* ante la salida de servicio de este.

La funcionalidad principal del protocolo prevé que cada nodo al iniciarse envíe un mensaje (*ALV)* al *master* (ver figura 1) para verificar su *existencia* (este mensaje puede ser realizado a través de un primitivo *ping* o a través de un paquete de datos definidos dentro programa de inicio). El protocolo prevé tiempos de espera E_n, antes de reenviar un nuevo mensaje *ALV*. En el caso de que el *master* se encuentre operativo (recepción de *ALV-ACK*) no se realizarán posteriores cambios durante el inicio del nodo. Por otro lado, si el mensaje enviado no obtiene su acuse de recibo, el nodo debe iniciar una secuencia de búsqueda de algún otro *master* sobre una lista *L* almacenada en un fichero auxiliar. La lista *L* mantiene en forma ordenada los números *IP* y nombres de *host* de cada nodo ($Nodo_0..Nodo_{N-1}$ para *N* nodos presentes en el cluster) en orden ascendente de acuerdo a su prioridad para convertirse en *master*. Esta lista es dinámica y es actualizada cada vez que se observa una contingencia en el cluster; lo que conlleva a mantener una copia actualizada de la misma en cada nodo.

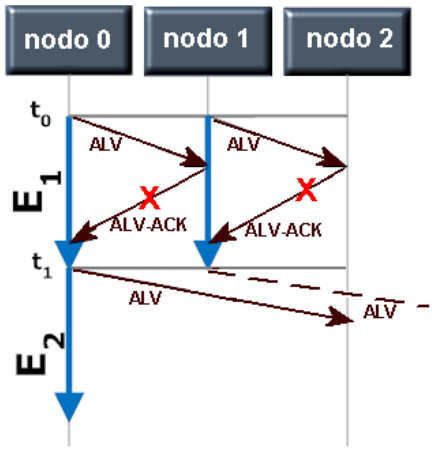

Fig 1. Intercambio de mensajes entre nodos desde el inicio del protocolo (t_0). En el ejemplo, ningún nodo recibe *ALV-ACK*.

La iniciación del cluster, generalmente tras un corte de energía, o simplemente tras una tarea de mantenimiento, prevé la iniciación de todos los nodos en forma simultánea. De esto se desprende que existe una probabilidad considerablemente alta de experimentar problemas de concurrencia en el momento que cada nodo busca la existencia del *master*. En particular, ningún nodo encuentra el *master* operativo y se inicia una búsqueda desordenada por parte de cada nodo. Esta situación obliga a establecer políticas de iniciación bien definidas dentro del protocolo. Luego, el mismo estará compuesto por una serie de funciones tendientes a garantizar una iniciación ordenada de los nodos.

Los próximos *workers* en iniciarse (Figura 1) detectarán un nuevo *master* operativo y se autoconfigurarán para responder ante éste como sus *workers*. (Eventualmente un *worker* puede haber cumplido el rol de *master* previo a la salida de servicio del cluster).

Ante la ausencia del *master*, cada usuario del mismo debe contar con un nuevo punto de ingreso al sistema (un nuevo *master*). Esta situación se resuelve al contar con un *host* servidor, externo al cluster, que presenta características de seguridad más robustas que los nodos del cluster en cuanto a software y hardware. Dicho servidor cuenta con información actualizada sobre la disposición de nodos *workers* y *master* en tiempo real. Luego, para el usuario final resulta transparente el ingreso al cluster a través de un potencial nuevo *master*.

Resultados y Objetivos

Los autores han creado un cluster de escala pequeña en un laboratorio de usos generales, con el objeto de probar la funcionalidad del protocolo *ISC*, y pequeños programas MPI [6]. Se encontraron buenos resultados en la reconfiguración de cada nodo al experimentar la iniciación de los mismos en diferente orden o en forma simultánea.

El trabajo futuro respecto de los resultados presentados corresponde a la formalización del protocolo, a través de su optimización y la incorporación de nuevas funcionalidades. El grupo de estudiantes muestra interés en dicha línea de investigación, observando también que existen dificultades de acceso a la información a soluciones computacionales similares.

Formación de Recursos Humanos

Esta línea de investigación prevé la formación de estudiantes (futuros ingenieros) en el área de computación paralela, en una relación estrecha con docentes investigadores que puedan brindar su asistencia y transmitir su experiencia.

De acuerdo al análisis de los próximos experimentos se evaluará la posibilidad de formulación de un proyecto referido a los temas acá mencionados.

Referencias

1. Morrison, R. Cluster Computing. Architectures, Operating Systems, Parallel Processing & Programming Languages. Documento Digital. r.morrison@ndy.com.
2. Hal Stem, Mike Eisler & Ricardo Labiaga. Managing NFS and NIS, 2nd Edition. O'Reilly. ISBN 1-56592-510-6.
3. Sehmeh, Klaus. Cryptography and Public Key Infrastructure. Wiley. ISBN 0 470 84745 X.
4. Crescitelli M., Bello E. & Monetti J. Construcción de un Cluster de Computadoras. Informe Interno Nº 1. Biblioteca Dpto. de Ingeniería en Sistemas de Información. Rodriguez 273 - Mendoza.
5. Bello E., Crescitelli M. & Monetti J. Protocolo ISC: Inicio Sincronizado de Nodos en un Cluster de Computadoras. Informe Interno Nº 2. Biblioteca Dpto. de Ingeniería en Sistemas de Información. Rodriguez 273 – Mendoza.
6. Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, Jack Dongarra. MPI: The Complete Reference. Documento digital disponible en: http://www.netlib.org/utk/papers/mpi-book/mpi-book.html